

RE Thomas.

**ATLAS
COMPUTER
LABORATORY**

**MULTI-
ACCESS
MANUAL**

**SCIENCE
RESEARCH
COUNCIL**

CHILTON, DIDCOT, BERKS

SCIENCE RESEARCH COUNCIL
ATLAS COMPUTER LABORATORY

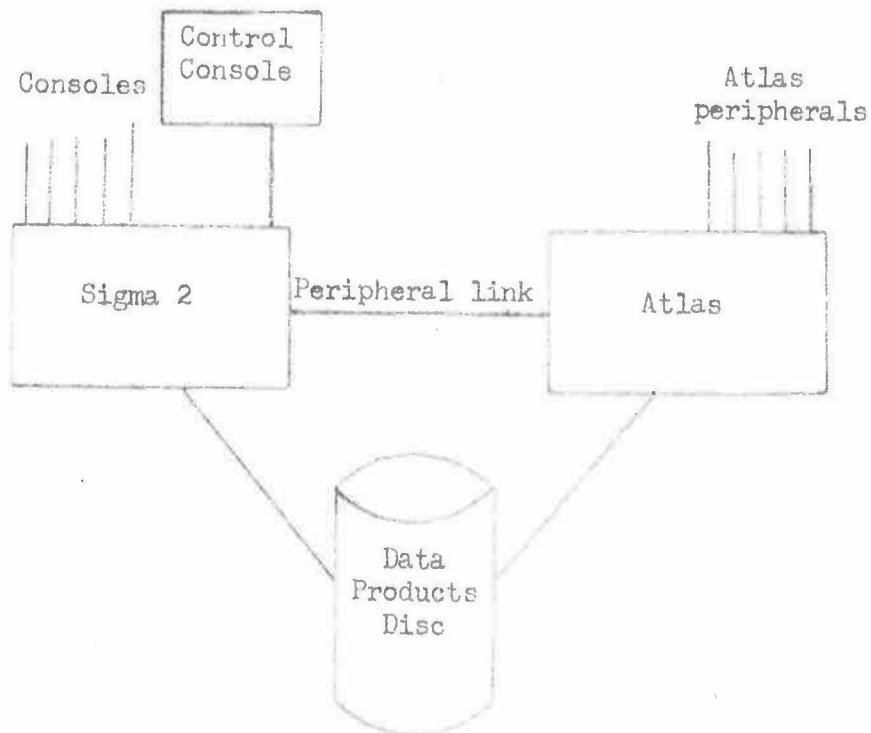
CHILTON MULTI-ACCESS SYSTEM

User's Manual

MAY 1969

Chilton Multi-Access System

The System provides the ability to create and manipulate files of information from remote consoles and to use these files for the initiation of jobs on Atlas. This file handling and console servicing are carried out by an SDS Sigma 2 computer using a Data Products disc (shared with Atlas) for storage of the files. No real-time interaction is possible with a job running on Atlas although output from the completed Atlas job may be directed back into the filing system.



Used for storage of files
and large-scale transfer of information
to and from Atlas

Schematic Configuration

The location of consoles and their numbers are included in the description of the WHO command (section 7).

CONTENTS

	SECTION
Use of the Console	1
Local Editing	2
Entry to the System	3
The Command Mode	4
Special Keys:	
The BREAK Key	5.1
The TAB Key	5.2
The Filing System	6
Command Syntax and list of Commands	7
Suggested Use of the Multi-Access System	8
Appendix I: AIC equivalents of teletype characters	

1. Use of the Console

The consoles are Teletype model 35, and for use with the System are operated as follows:-

1. Plug in and switch 'ON' at wall socket. The red control panel light marked 'MAINS' will illuminate.

If Sigma 2 is also switched on (though not necessarily running the System) the green MODEM READY light will also illuminate.

2. Turn the right-hand round selector knob to ON-LINE.
3. Depress the extreme right-hand green START button on the control panel. This will illuminate and remain depressed and the teletype motors will start.
4. For all normal working the left-hand round selector knob should be switched to K. However, if a paper tape copy of all activity is required, use the KT position.

The console normally operates in DUPLEX mode. This means that when a key is depressed, a signal is sent to Sigma 2 but no printing occurs. The printing mechanism is activated by transmission back from Sigma 2; this echoing gives the user some confidence that the printed character has actually been received by the System.

Echoing is suppressed in the following circumstances:-

1. during input for a PASSWORD or KEY in order to maintain security (in this case it will be resumed by a closing bracket or carriage return);
2. when an illegal character is received;
3. when input of that character is not expected.

Input is accepted from the console line by line, a line being terminated by the carriage return (CR) key. This character is echoed by both CR and line feed (LF), effectively changing it to a newline (nl) character. The LF key on the console is ignored.

If a user possesses a console which cannot operate in DUPLEX mode, that console can be selected in HALF DUPLEX (or SIMPLEX) mode by the System operators. In this mode, printing occurs as soon as a key is pressed. A PASSWORD or a KEY will be requested by a series of eight 'splodges', over which the user may type. The disadvantages of half duplex working are

1. Only eight characters of a KEY or PASSWORD will be unreadable.
2. Attempting to type characters while output is in progress will cause garbled characters to be printed.

3. The 'duplication' facility is not available (see section 2).
4. A line containing 82 characters will be terminated in the manner specified in section 2, but the replaced character, although ignored, will be printed.

If a breakdown occurs in the cabling between the console and Sigma 2, the red CAR FAIL light will illuminate on the control panel and an alarm will be sounded in the teletype. The System operators should be informed of such an occurrence.

After logging out of the system (for details see the FINISH command, section 7), ensure that the console is switched off as follows:-

1. Press the green START button on the control panel a second time. It will then be released and the motors will stop.
2. SWITCH OFF THE MAINS SUPPLY AT THE WALL SOCKET (or, if no switch exists, remove the plug).

A console should not be left switched on and unattended for periods of more than a few minutes. The red MAINS control panel light indicates whether power is on or off.

Console Paper Tape Reader

Currently, use of the paper tape reader should be avoided.

2. Local Editing

NO ACTION IS TAKEN BY THE SYSTEM on any part of a line UNITL
(nl) IS RECEIVED.

The user is able to specify two characters which will give some local editing facilities within a line (see LOCALDC and LOCALDL command specifications), i.e. before the newline character is used. These settings will be preserved from one session to another. On initial entry to the system, they are set as ← and ↑ respectively. The first of these (←) causes the preceding character to be deleted from the input line, e.g.

```
both          ABCE←DE
and           ABDE←←CDE
are treated as ABCDE
```

The second (↑) causes the whole line to be ignored and after echoing is followed by newline.

When a user types his identification number at log-in time (see section 3), the standard characters (←↑) will achieve any necessary local editing, irrespective of the settings chosen by that user.

Use may be made of local editing characters not only when commands are taken from a console but also from a file (see definition of the APPLY command, section 7).

A further facility is available which roughly corresponds to the 'DUP' button on the IBM 026 card punch. This facility is not available to users whose consoles operate in HALF DUPLEX (or SIMPLEX) mode. On pressing CONTROL and F simultaneously (the RU character), the previous content of the input buffer at that position is resurrected. Thus, for example, using φ for the RU character:-

```
typed characters (^ = space) . line input to the System
line 1      T H E^ C A T^ S A T^ D O W Nnl)  T H E^ C A T^ S A T^ D O W Nnl)
line 2      φ φ φφ D O Gnl)                T H E^ D O Gnl)
line 3      O N φφ φ φ φ^ φ φ φφ U Pnl)    O N E^ D O G^ S A T^ U Pnl)
```

(The commands APPLY and RUN overwrite the contents of the input buffer and use of the RU key immediately following these commands will give unexpected results.)

A line (including the newline character) may not exceed 82 characters, and, irrespective of the character typed, newline is inserted at the 82nd 'column'. In HALF DUPLEX (or SIMPLEX) mode this last character, although ignored, will be printed.

Any illegal character encountered during input will be echoed with

****FAULT CHARACTER

and the acceptable current contents of the buffer are output ready for the user to continue typing.

3. Entry to the System

A potential user of the System must obtain a user identification number ('userid') before access is permitted.

To initiate the log-in procedure, the CR key is pressed. At this stage all other characters are ignored. If there is no response at all to CR and the console has been switched on correctly (section 1), then the System is not running.

If too many people are currently logged in, the response is

***SYSTEM FULL

and further access is impossible.

When the System is about to close down, the message

***END OF SESSION

will appear, again preventing access.

Otherwise, the System responds with

USER nl ^^ (^ = space)

Normally, whenever input is expected, two spaces are output at the beginning of a line. This aids subsequent recognition of whether the user or the System typed a given line. A pointer on the type box (model 35 Teletype) indicates the next typing position.

The user now types his 'userid' and if this is unacceptable provokes the response

UNKNOWN USER

(see example 3.1) and the log-in procedure must then be recommenced. Corresponding to each 'userid' there is a PASSWORD which is requested by

^^ (

Echo checking is then inhibited until a terminating nl or) nl is received and the intervening characters are treated as the user's password. Up to three attempts are allowed (example 3.2) for the input of this password after which the log-in procedure must be restarted. Initially a user's password is null so that nl or) nl should be typed and having thus gained access he can set it to the required character string by the PASSWORD command (q.v.). If a password is forgotten it is possible for it to be reset to null.

An acceptable password provokes the message

STARTED AT [time] ON [date]

(see example 3.4), which may be followed by a "message of the day".

At this point the user moves into COMMAND mode.

USER
BLOGGS
UNKNOWN USER

Example 3.1

USER
I0000
()
ILLEGAL PASSWORD
()
ILLEGAL PASSWORD
()
TOO MANY ATTEMPTS

Example 3.2

USER
I0000
()
USER NUMBER CURRENTLY IN USE

Example 3.3

USER
I0000
()

STARTED AT 15.11.27 ON 23/04/69

Example 3.4

HERE FOLLOWS, PERHAPS, A ONE LINE 'MESSAGE OF THE DAY'

4. The Command Mode

In this mode, commands may be input to the System for the manipulation of files and the running of jobs on Atlas. These commands will usually be typed on the console. Alternatively, they may be taken from a specified file (for which the phrase "during an APPLY" will be used). For details see the APPLY command, section 7.

Whenever a user enters the command mode, either for the first time or after obeying a command, certain warning messages may be typed by the System:-

***MESSAGE FOR YOU

indicates that some user has made an entry in your message file. It is not essential that action be taken at this stage. For complete details see the description of the MESSAGE and SIGNAL commands.

***DISENGAGED

indicates that the console has been de-coupled from the System for management reasons and no further input will be accepted.

***ENGAGED

at a later stage signifies the re-coupling of the console and re-entry into the command mode.

***ATLAS JOB COMPLETED

informs the user that an Atlas job initiated by the RUN command (q.v.) has been completed. Nevertheless, there may still be some delay before output destined for files actually appears in the filing system.

***ATLAS JOB LOST

will be seen if the job has been lost in Atlas before it has been able to execute (eg in the case of a 'restart').

***APPLY TERMINATED ON LINE n

indicates that an error has occurred during an APPLY, and no more commands will be taken from the file. The command causing the error is at line number n (see section 6 re line numbers). This message may occur during the use of a SYSTEM file (see section 6). Further commands are then expected from the console.

Special instructions may be issued from time to time by the operator and these should be obeyed immediately. Such messages will be preceded by ***.

Other warning messages may be output at any time during a console session:-

***BREAK? -

indicates that the break key has been pressed. The BREAK will take full effect if a Y or a * is typed. (See section 5.1).

***FILE SYSTEM FULL

indicates that there are no more disc blocks available for that user. This will occur during the various commands (see command descriptions) which involve the creation or expansion of files. The user should TYPE such files to see what state they are in before using them in any other manner.

***TO REMAIN IN CURRENT STATE, PRESS THE RETURN KEY WITHIN ONE MINUTE.
CURRENT INPUT LINE READS

This message will be followed by the current input line typed so far by the user. It indicates that the System is waiting for a line of input, and that the user has not pressed newline during the last four minutes. If newline is not pressed within a minute after the message has been typed, the current line will be lost, and the System will react as if

***Z

had been typed. If, however, the user was in command mode at the time, he will be logged out.

***LOGGED OUT BY SYSTEM

This will occur after a user has been forcibly FINISHED by the System. This message will appear rarely unless the user has logged in on the automatic modem line, for which there is a time limit.

5. Special Keys

5.1 The BREAK Key

Each console has a BREAK key (situated on the right-hand side of the keyboard) which may be used to interrupt certain commands. When used, any output or message currently in progress is terminated and the message

***BREAK? -

is output. Typing Y or * (the first non-space character only is taken) causes control to pass back to the command mode.

Any other reply effectively revokes the 'break' (although some output is always lost) and the System will continue to complete the interrupted command. A confirmed 'break' during an APPLY will prevent any further commands being taken from the specified file. In addition,

ON LINE n

will be typed, indicating the line number (n) of the command being obeyed when the BREAK key was pressed (see section 6 re line numbers).

The following commands only are interruptable by BREAK:

TYPE, MESSAGE, WHO, EXAMINE, EDIT

otherwise, the BREAK will not be acknowledged until return to command mode. However, the BREAK key is treated somewhat differently in EDIT: for further details see page 7.Edit.7.

5.2 The TAB key

There is a TAB key (CONTROL and I) on the console which is treated specially by the System. By using the TABS command (see section 7) the user can set tab positions which are preserved from one session to the next. Subsequent use of the TAB key will cause the carriage to space forwards to the nearest specified position, and spaces to be inserted into the input buffer. The initial settings are at positions 8, 16, 24 etc. If the console is working in half-duplex mode, however, the carriage will not move to the correct place. On a Teletype 35 in half-duplex mode, which has a mechanical tab mechanism, the carriage will move to the next preset mechanical tab position. On a Teletype 33, the carriage will not move at all. The correct number of spaces will, however, be inserted in the input buffer and subsequent printing will be correct.

Note that the above facility is similar to the SKIP button on a card punch when using a drum card. It is NOT like the tab on a Flexowriter which guarantees at least two spaces.

6. The Filing System

A file consists of a collection of records, each terminated by nl and preceded by a line number. A record consists of ≤ 82 of the legal characters obtainable at the console; binary records are not allowed. Currently up to 95 files may be owned by any one user and the only restriction on file length is that imposed by the maximum value (32767) of a line number.

Each file has a file name associated with it. This name consists of two parts, separated by /. The first part is the userid of the creator of the file, and the second is a unique identifying name of up to twelve characters, chosen by the creator. Space, comma, or / may not be used in either part. When the creator of a file wishes to access it, the first part and the / may be omitted.

Files are stored on the Data Products Disc. As disc space is at a premium, a form of file rationing operates. Files may be selected either as temporary or permanent. Each user will be given a ration of permanent file space, which may not be exceeded. He may, however, create any number of temporary files. In general, temporary files older than one month will be deleted at the beginning of each month. If a user exceeds his permanent file space allocation either when creating a new file, increasing the length of an old file, or by changing the status of a file (see below), that file will be allocated temporary status.

In order to safeguard the security of the contents of a file, three user groups have been defined:-

- (i) the file owner (i.e., the person who created it);
- (ii) a keyholder - a person able to quote the key (or password) set for that file by the owner;
- (iii) the public (i.e., everyone other than the owner). Keyholders are a subset of the public.

For each user group, the file owner will grant a status, or degree of access. The status is set at the time of creation but may be altered subsequently with the STATUS command. For convenience, the file status also includes information as to whether the file is to be considered temporary or permanent.

The following degrees of access are available:

- (i) NULL i.e., no access whatsoever;
- (ii) READ implying that any attempt to write into the file is to be prevented;
- (iii) APPEND which gives a user the ability to make additions to the end of the file, but not to delete it;
- (iv) FREE giving complete access.

The expression 'write status' will be used for either append or free access.

A status is specified in the form

xyz.q

where x is the status of the owner

y is the status of keyholders

z is the status of the public

x, y, and z must be one of the letters N, R, A, F according to the desired access.

q = T for a temporary file

P for a permanent file

A valid status might then be

ANN.T

When a user requests access to a file, his status is checked in the manner shown in Fig. 6.1. A key is requested in an exactly similar way as a password.

Whenever the contents of a file are accessed the usage count for the appropriate user group is incremented.

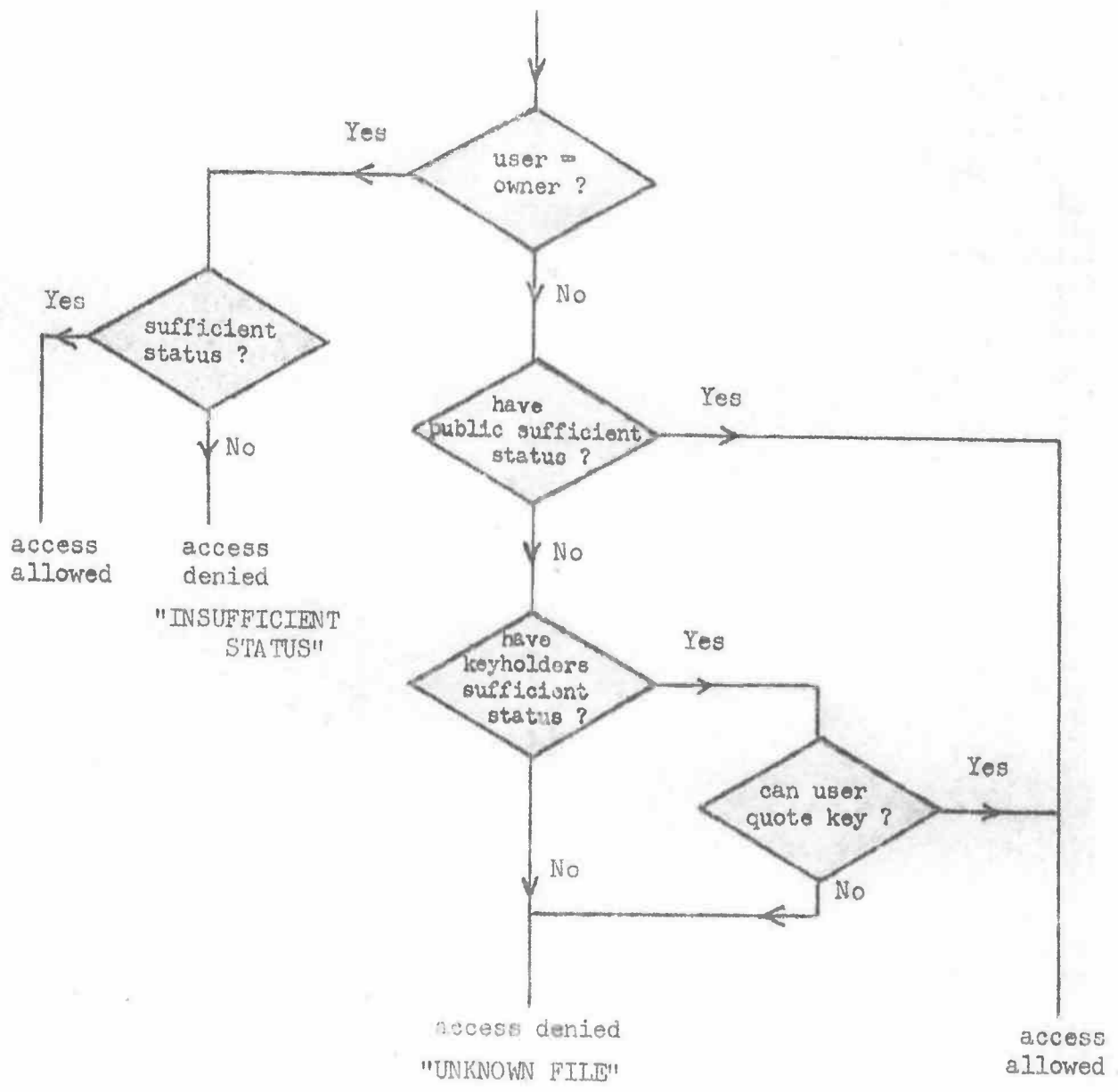
The simultaneous reading and writing of a file is prevented by the System.

Two further points should be made about files. Firstly, unlike some multi-access systems, there is no concept of a 'current file'. Thus filing commands will always require at least one file name as an argument. Furthermore, a user is protected from destroying his files by overwriting them since the deletion of a file can only be achieved by explicit use of the DELETE command.

Section 7 lists alphabetically the commands available and their effect.

Normally, no significance is attached to the contents of a file. However, there are two exceptions:

- (i) files may contain sequences of commands, possibly using parameters. A full description appears under the APPLY command (section 7.Apply).
- (ii) Atlas jobs may be initiated by nominating one particular file to be input to Atlas. This file will start with an Atlas job description which may refer to other system files as input or output streams. Details are given under the RUN command (section 7.Run).



File Access Validity Checking

Figure 6.1

System Files

There is a library of files available to every user. These files are owned by a user called SYSTEM. They may be addressed either by their full title, e.g. SYSTEM/FILE1, or with the word SYSTEM omitted, e.g. /FILE1. Some of these files contain sequences of commands which may be applied by the user. An example of this is the file /HELP. This may be applied either by specifically stating the APPLY command, e.g.

APPLY /HELP

or by omitting the word APPLY (in which case the word SYSTEM must also be omitted), e.g.

/HELP

Thus any command line beginning with / is treated as an APPLY for the specified SYSTEM file. Note that any parameters required must be given in the format corresponding to APPLY (see command definitions).

SYSTEM files currently existing are:

<u>Name</u>	<u>Example of Use</u>	<u>Content</u>
/HELP	/HELP	Short synopsis of all command definitions
/LINEDIT	/LINEDIT,filename	Sequence of commands to line edit a file, delete the old file, and rename the new file with the previous name. This 'pseudo-command' will fail unless filename belongs to the user.
/NEWS	/NEWS	Notes about recent innovations to the System.
/STANDARD	/STANDARD	Current maxima of parameters permitted in Atlas job descriptions of STANDARD Atlas jobs.
/EDIT	/EDIT,filename	Sequence of commands to context edit a file, delete the old file, and rename the new file with the previous name. This 'pseudo-command' will fail unless filename belongs to the user.
/WHOSWHO	/WHOSWHO	A list of users and job numbers known to the System.
/EXAMINE	/EXAMINE,file1,file2,...	Examines separately all the named files.
/LIST	RUN /LIST,userid and name,file1	Runs an ATLAS job to list files on the lineprinter.
/MERGE	/MERGE,file1,file2,file3	Creates file3 by merging selected sections from file1 and file2 (neither of which is altered in the process).
/SEARCH	/SEARCH,file1	edits file1 into XJUNKX which is then deleted. This is very useful for doing table searches (eg - finding out who a user is from /WHOSWHO without typing out the whole file).
/:	/:,command1,command2,...	a device for setting up a series of commands to be obeyed sequentially without the need to create an Apply file. The author finds this useful for such sequences as:-

/:,TYPE CRUDFILE,EXAMINE,FINISH

while he goes off for a cup of coffee!

<u>Name</u>	<u>Example of Use</u>	<u>Content</u>
/ARCHIVE	/ARCHIVE, filename, type, tapename, blockno, userid and name	Causes an Atlas job to be run which copies filename to a tape or disc called tapename from block blockno onwards. type should be the word TAPE or DISC to indicate which medium is being used. OUTPUT 0 will indicate the position of the last block of the file.
/RECOVER	/RECOVER, filename, type, tapename, blockno, userid and name	Performs the reverse of the above. The information on TAPE or DISC tapename, at block blockno, is copied into filename (or appended to it if it exists already).
/RETRIEVE	/RETRIEVE, word, userid and name	Performs an information retrieval, depending on word, and puts the results into a file called PAPERS. word should be either KEY or AUTHOR. If the userid and name is followed by LSE, retrieval takes place from the LSE tape rather than the standard one. (The London School of Economics have a special tape of information on statistical journals.)
/OPERATOR	/OPERATOR, userid and name	This should be used only by someone remote from the Chilton site who is unable to contact an operator. It will run an Atlas job which will bring his plight to the notice of the operators in the Atlas room.
/BOOKING	TYPE /BOOKING EXPAND /BOOKING	Remote users can TYPE this file to see the current booked periods on the manual modem, and can EXPAND it to add requests for periods in the next week.

7. Command Syntax

The purpose of this section is to define the terms in the descriptions of the various commands.

A [COMMAND] is a sequence of alphabetic characters terminated by space or newline. Depending upon the context, sometimes space is the only valid terminator (e.g. DELETE which must be followed by the name of the file to be deleted).

A [CHARACTER] is a single symbol which may be printed on the console keyboard.

A [FILENAME] consists of two parts, separated by /. The first part is the userid of the owner, i.e. the person who created that file, and the second part is a unique identifying name of up to twelve characters. The first part and the / may be omitted whenever a user is accessing his own files.

A [LINENUMBER] is a decimal integer in the range 0 to 32767; an [INCREMENT] is defined similarly.

An [ARGUMENT] is either a [FILENAME], a [LINENUMBER], a [CHARACTER], or an [INCREMENT]. Each of these may be terminated by space, comma, or newline.

A [QUALIFIER] is a collection of alphabetic characters terminated by a space or newline.

An [ARGUMENT PHRASE] consists of either a [QUALIFIER] or a comma followed by an [ARGUMENT] (the exception being the [QUALIFIER] NUMBERED which has no [ARGUMENT]). A command needing several arguments may have [ARGUMENT PHRASE]s presented in any order only if a [QUALIFIER] is used instead of a comma; otherwise a fixed order is required. Thus

```
TYPE MYFILE FROM 1 TO 50
```

```
TYPE MYFILE TO 50 FROM 1
```

```
TYPE MYFILE,1,50
```

gave identical results, different from that obtained with

```
TYPE MYFILE,50,1
```

SUMMARY OF COMMANDS

(A full description appears on page number 7.Commandname)

ADDTABS	adds extra tab settings
APPEND	concatenates two files
APPLY	uses a file as source of commands
COPY	copies a file into a new one
CREATE	creates a new file
DELETE	deletes a file
EDIT	creates a new file by context editing an existing file
ENDAPPLY	stops using the file as a source of commands
EXAMINE	types contents of user's file directory
EXPAND	lengthens existing file
FINISH	ends a console session
KEY	sets a key for a file
LASTLINE	types the last line number of a file
LINEDIT	creates a new file by line editing an existing file
LOCALDC	sets a new character for 'delete last character' symbol
LOCALDL	sets a new character for 'delete current line' symbol
LOCALTAB	sets a character as tab symbol for DIDS display
LOGOUT	ends a console session
LOSETAB	removes specified tab settings
MESSAGE	lists any messages for the user
PASSWORD	sets a new password
RENAME	changes the name of a file
RENUMBER	changes the line numbers of a file
RUN	initiates a job in ATLAS
SIGNAL	sends a message to a user
STATUS	sets a status for a file
SUBMIT	sends non i/o file jobs to ATLAS
TABS	sets up a new list of tab settings
TYPE	lists a file on the console
UNVERIFY	prevents commands being listed during an APPLY
VERIFY	allows commands to be listed during an APPLY
WHO	lists all users currently logged in

Standard Format

ADDTABS number1 AND number2 AND

Purpose

To add more horizontal tab settings to those already set. Any number of settings may be specified and each number (which must be less than 80) is terminated either by one space, a comma, or (for the last) by newline.

Normal Response

The above command is similar to the TABS command, except that the current tab settings are not cleared prior to the inclusion of the new settings. Numbers may be specified in any order. The positions past the last number specified are not set as TAB locations.

Error Response

ERROR IN TAB SETTING	typing error in one of the arguments, or a setting beyond position 79 has been specified. All tab settings up to the wrong one will have been accepted.
----------------------	---

Notes

If tab position 0 is added, no conversion of the tab character into spaces will take place, but the tab character will be inserted in the file, and the mechanical tab mechanism of the console (if any) will be activated when the file is TYPED. . Special care must be taken, since an unconverted tab will not match spaces in a string in EDIT, and Atlas will not accept the character in any input file.

Standard Format

ADDTABS number1 AND number2 AND

Purpose

To add more horizontal tab settings to those already set. Any number of settings may be specified and each number (which must be less than 80) is terminated either by one space, a comma, or (for the last) by newline.

Normal Response

The above command is similar to the TABS command, except that the current tab settings are not cleared prior to the inclusion of the new settings. Numbers may be specified in any order. The positions past the last number specified are not set as TAB locations.

Error Response

ERROR IN TAB SETTING typing error in one of the arguments, or
a setting beyond position 79 has been
specified. All tab settings up to the
wrong one will have been accepted.

Notes

If tab position 0 is added, no conversion of the tab character into spaces will take place, but the tab character will be inserted in the file, and the mechanical tab mechanism of the console (if any) will be activated when the file is TYPED. . Special care must be taken, since an unconverted tab will not match spaces in a string in EDIT, and Atlas will not accept the character in any input file.

Standard Format

APPEND filename1 TO filename2

Purpose

To add filename1 to the end of filename2. The user requires read status for filename1 and write status for filename2. Any necessary keys are requested in the normal way with no indication as to which file requires the key (the user should know!). If both require keys, that for filename2 is requested before that for filename1.

Error Responses

All file errors refer either to filename1 or filename2.

ERROR IN FILE TITLE

typing error in a filename.

UNKNOWN FILE

file does not exist or an incorrect key has been specified.

INSUFFICIENT STATUS

user does not have the correct status for one of the files.

ERROR IN QUALIFIER

typing error in TO

FILE BEING USED

one of the files is being accessed by someone else.

***FILE TOO LONG

a line number which is too large has been generated. Only part of filename1 will have been appended.

PERMANENT FILE SPACE EXCEEDED - FILE GIVEN TEMP. STATUS

filename2, originally a permanent file, is now too long to remain so, and has been reset as temporary.

Note

The increments found in filename1 are preserved in the APPEND. The first number is taken to be the sum of the last number of filename2 and the first number of filename1. Thus, if filename1 has line numbers 10,20,30 and filename2 has numbers 1,2,3, the new filename2 will be numbered.

1,2,3,13,23,33

Standard Format

APPLY filename USING parameter1 AND parameter2 AND ... FROM
startnumber

Alternative Format

USING and AND are interchangeable qualifiers. The word APPLY may
be replaced by : as in

:filename USING parameter1 AND parameter2 AND ... FROM startnumber

Default settings

Any number of actual parameters (including none) may be specified
according to the number of formal parameters used in filename (see
below for details). Each parameter may have either USING or AND as
qualifier. In the above context, FROM may not be replaced by a
comma but must be explicitly stated (if required). The default setting
is the first line of the file.

Purpose

To take commands from the specified file instead of the console.
Only the first input line, containing the command itself, is taken from
filename, and any other input for that command is expected from the
console. For example, the CREATE command will request the status, key,
and file contents from the console rather than taking them from the
file itself.

APPLY commands may not be nested. (Note that /EDIT for example is
an implied APPLY command - see section 6.)

Normal Response

Commands are taken from the file beginning at line startnumber.
If no such line exists, the line with the next highest number is taken.
If the user requires keyholder status, a key is requested in the normal
way.

Error Responses

ERROR IN FILE TITLE	typing error in filename
UNKNOWN FILE	file does not exist, or user has specified an incorrect key.
COMMAND ILLEGAL DURING APPLY	APPLY commands may not be nested.

FILE BEING USED	someone else is writing to filename
ERROR IN FORMAL PARAMETER	see below for details
INSUFFICIENT STATUS	user does not have read status for this file
FILE TOO SHORT	startnumber is greater than the last line number in the file

Notes

As each command is reached, the command line is printed on the console. The command

UNVERIFY

within the APPLY file will suppress this printing. This command is cancelled by

VERIFY

Both these commands are only meaningful within an APPLY file.

Formal parameters may be specified in filename by numbers surrounded by <>, thus: <1> <2> <3> The numbers refer to the order in which the actual parameters are specified on the command line of APPLY. When the line containing one or more formal parameters is taken as a command line, the formal parameters are replaced by the actual parameters before any printing takes place.

Consider the example:

```
APPLY myname USING filename1 AND filename2
```

If myname has the form:

```
RENAME <2> AS <1>
TYPE <1> NUMBERED
```

then the commands obeyed will be

```
RENAME filename2 AS filename1
TYPE filename1 NUMBERED
```

An APPLY is terminated either on filename being exhausted or on meeting the command

```
ENDAPPLY
```

A premature termination may be achieved by the use of the BREAK key.

Input of parameters

If the command line is imagined to consist of sets of strings of characters separated by qualifiers, then parameters are synonymous with those strings. Qualifiers consist of AND, USING, or a comma, so in order to be able to include any of these (or the word FROM) within a parameter string, a pair of pointed brackets < > must enclose the whole parameter.

Thus for example in the following command line:-

APPLY BILL USING CHARLIE AND DICK, ERIC THOMAS, <FRED AND CO,>

we have:-

PARAMETER	<0>	<1>	<2>	<3>	<4>
STRING	BILL	CHARLIE	DICK	ERIC THOMAS	FRED AND CO,

Note in the above example that the apply file BILL becomes parameter <0>, also that parameter <4> is allowed to contain qualifiers as pointed brackets surround it.

Note that multiple spaces before, and a single space after the words USING or AND get ignored. This does not apply to the comma alternative.

A maximum of 40 parameters is permitted, and further items are ignored.

Null parameters may be specified by using the comma alternative to the qualifiers, and placing two commas adjacent (or separated by a space).

Editing in the Command Line

The characters chosen for local editing (see section 2 and definition of LOCALDC, LOCALDL commands) retain their editing significance if encountered in a command line. These characters must of course be included in the file before they are selected for local editing. This enables a user to vary the effect of a given file merely by selecting different local edit characters. Consider the following file:-

```
CREATE FRED  ↑ ←
```

```
DELETE FRED  @ $
```

If we have selected ↑ as 'delete line' and \$ as 'delete character', then the above will be equivalent to

```
DELETE FRED
```

If however we select @ as 'delete line' and ← as 'delete character', then the original file is equivalent to

```
CREATE FRED
```

A command line is assembled character by character. The user should therefore note that

1. The 'delete line' character will always delete a line even if there are other local edit characters after it.
2. Commands such as DELETE which have a fixed number of arguments may have any number of other characters after the last argument without preventing correct execution of the command. Thus, if \$ was not selected as 'delete character' in the first case above, the command line

DELETE FRED @ \$

would be printed, the effect being the same as before. However, if in the second case ← was not selected as 'delete character', the line printed would be

CREATE FRED ↑ ←

and would lead to the message

ERROR IN QUALIFIER

since the qualifiers FROM and STEP may appear after this command.

3. Any local edit character appearing between < and > of a formal parameter will cause the error:

ERROR IN FORMAL PARAMETER

Any 'delete character' symbols appearing immediately after the > will cause the deletion of characters in the corresponding actual parameter.

4. Any formal parameters specified before a 'delete line' character in a given line must be correctly satisfied (ie, a corresponding actual parameter must exist).

Standard Format

COPY filename1 INTO filename2

Alternative Format

COPY filename1 GIVING filename2

Purpose

To produce an exact copy of filename1, with the name filename2 .
This command is equivalent to using

LINEDIT filename1 INTO filename2

and immediately typing ***Z after ***A. All messages that can
appear here are a subset of those described under LINEDIT.
(No ***A is output.)

Standard Format

CREATE filename FROM startnumber STEP increment

Default Settings

startnumber: 10

increment: 10

Purpose

To enter filename in the user's directory and to associate subsequent input with this filename. A user may create files only for himself.

A line number is associated with each line of a file. The first line is given the startnumber and subsequent lines are numbered according to the specified increment. Thus a standard file will be numbered 10,20,30,40... .

Normal Response

The status required for this file is requested by the message

STATUS?

The expected reply is a legal status (for details see Section 6). If a keyholder is given anything other than null status, a key is requested by the message

KEY?

^(

The expected reply is a legal key (for details, see KEY command).

Following these messages,

***A

is output, indicating that subsequent input is to be taken as the file specified.

The file is terminated by the user typing

***Z

by itself on a new line.

Error Responses

After the command line, the following error messages may appear:

ERROR IN FILE TITLE	typing error in filename
FILE EXISTS ALREADY	filename has already been used.
ERROR IN QUALIFIER	typing error in FROM or STEP
ERROR IN LINE NUMBER	typing error in startnumber or increment. Max. number size is 32767.

After STATUS? message:

STANDARD STATUS ANN.T ASSUMED	illegal status specified.
TOO MANY FILES	user already has 95 files.
FILE SYSTEM FULL	all space on the disc has been allocated. There is no room to create a new directory for the user, or there is no room for the first block of the file.

After ***A,

***FILE TOO LONG	a line number greater than 32767 has been allocated. The file is terminated and the last typed line is ignored.
FILE SYSTEM FULL	all the current available disc space has been allocated. The file is terminated with a certain amount of typed input ignored (depending on where a new block has been requested).
PERMANENT FILE SPACE EXCEEDED - FILE GIVEN TEMP. STATUS	filename, originally selected as permanent, is too long to remain so, and has been reset as temporary.

Standard Format

DELETE filename

Purpose

To remove a file from the filing system and to clear the entry in the user's file directory. Any necessary key is requested in the normal way. Free access is necessary for deletion of a file.

Normal Response

FILE DELETED

Error Responses

FILE BEING USED

a file cannot be deleted whilst it is in use.

UNKNOWN FILE

the file does not exist or a wrong key has been given.

ERROR IN FILE TITLE

typing error in filename

INSUFFICIENT STATUS

free access is required.

Standard Format

EDIT filename1 GIVING filename2

Alternative Format

EDIT filename1 INTO filename2

Purpose

This command creates a new file, filename2, as an edited version of filename1, retaining the status of filename1, but having null key. Any necessary key for filename1 (read status is required) is requested in the normal way.

Textual alterations to filename1 are made by reference either to their context or to their line number.

Normal Response

***A

indicates that input can proceed.

Character strings (which must not include newlines) are used to locate the position at which alterations to filename1 are required. A character string is 'delimited' by any convenient character, with the following restrictions:

- (i) it must not occur in the string (obviously);
- (ii) it must be a non-alphanumeric printing character (i.e. not space, tab, digit, letter, etc.)

Conventionally we will write the delimiter as ϕ . It may, of course, vary from editing command to editing command, and will be taken as the first printing character following the appropriate command letter. The character string is then taken as all the characters between (but not including) this delimiter and its next occurrence. Thus the command

L ϕ abcd ϕ

will cause the editor to search for the character string abcd in filename1. If convenient, a delimiter at the end of a line may be omitted.

All editing is carried out with reference to a current line and a (notional) character pointer. The current line may be selected either by search commands or by M and N commands (which are all described below). Having selected a line of the file, the editing commands refer to this line.

Commands may appear singly on a line, or grouped together. When grouped, the character pointer is moved progressively along the current line of the file and no string matching is attempted behind it. The pointer is reset to the start of a line whenever a newline is encountered, either by meeting the end of the command line or by meeting a newline in filename!. When a line of commands has been obeyed, EDIT outputs a newline indicating that further console input is required.

Any numbers (as opposed to character strings) following a command must be terminated by at least one space.

Modes of Operation

The user can choose one of two modes which determine whether or not intervening text is copied or deleted during a search command.

- C Set Copy mode. All intervening text will be copied.
- D Set Delete mode. All intervening text will be deleted.

C is assumed initially. D must be followed by a space or newline, giving some safeguard against selecting this mode by accident.

The following two commands act independently of the mode selected.

- Mn Miss out the rest of the current line (i.e., forwards from the character pointer) and a further n lines.
- Nn Copy the whole of the current line and the Next n lines.

If n is omitted it is assumed to be zero and only the current line is affected. The line (n + 1) beyond the present one is selected as the new current line.

Search Commands

Two basic search commands are used to select a line to be the new current line of the source file. Searches will always start from the present position of the character pointer.

- F ϕ S1 ϕ Find the line beginning with character string S1 and take this as the current line. The search will begin at the present line only if the character pointer has been reset to the start of it. If S1 is null, the next completely blank line will be found (i.e. a line without even a space in it).
- Fn Find the line whose line number is n or the nearest line number greater than n. If n is less than (or equal to) the current line number then no action is taken.

L ϕ S1 ϕ Locate the line containing the string S1 (i.e., the appearance of S1 anywhere from the present position forwards within filename1) and take this as the current line. If S1 is null, a match will be found at the present position.

If D mode has been selected, all the material in filename1 from the present character pointer position up to but not including S1 is deleted. If the character pointer is in the middle of a line at the start of the search, the newline character will be retained for that line.

Failure to find a match for the string S1 will exhaust filename1. If this occurs the user must return to the command language (by typing ***Z) and re-enter EDIT, this time using filename2 as the source file.

Consider the following example (* indicates the current line).

Section of source file:	* this is a small test of the L command
Command line:	L ϕ sm ϕ D L ϕ co ϕ C
Corresponding section of edited file:	this is a * command

The character pointer will be left at the beginning of the specified string.

Amending Commands

These commands perform the actual editing and reference strings in the current line only. Note that only one delimiter is required between strings S1 and S2. If S1 is null, a match will be found at the present position.

A ϕ S1 ϕ S2 ϕ	Find string S1 in the current line and insert string S2 After it.
B ϕ S1 ϕ S2 ϕ	Find string S1 in the current line and insert string S2 Before it.
E ϕ S1 ϕ S2 ϕ	Find string S1 in the current line and Exchange it for string S2. If S2 is null, S1 will be deleted.

An amendment will not be carried out if the resulting line would exceed 82 characters (an error message is given). A, B or E commands may not be used if D mode has been selected.

These commands only affect the first occurrence of string S1 and when completed the character pointer is positioned immediately after string S2.

Consider the following example.

Line of source file:	the cat sat on the mat
Command line:	A ϕ the ϕ big ϕ B ϕ on ϕ down ϕ E ϕ the ϕ a ϕ
Edited line:	the big cat sat down on a mat

(Note the space after down and the space before big in the command line)

Moving the Character Pointer

In addition to the methods described above, we have the command

Pn Position the character Pointer at "column" n
 in the current line.

If n exceeds the length of the line, the pointer is placed on the **last** character. No copying or deleting is implied when the pointer is moved with this command.

Verification

It is possible to check that the desired amendments have been carried out.

V Set Verify mode. When the command line has been obeyed, the current line will be typed out.

U Unset verify mode. No typing will occur upon completion of the command line.

Initially V is assumed, and the typed line is preceded by the line number terminated by one space. This line number and space cannot be altered by EDIT. The character pointer position is always taken relative to the start of the text proper. Thus, if a verified line reads

```
10^JOB^I2000^YOURNAME^ETC
```

the command

```
P5
```

will position the character pointer at the character I

Global Commands

A global command is one which is obeyed repetitively, e.g. 'alter every occurrence of S1'

G Obey the rest of the command line repetitively

The repetition will continue until a search is unsuccessful in F,L,M,N,A,B, or E commands. In the case of F,L,M, and N commands, filename1 will have been exhausted, and in these cases no error message is generated. Any other error will cause termination of the sequence.

Only one G command may appear on a line, i.e., global commands cannot be nested.

Consider the following examples.

- (a) Alteration of all occurrences of) in the first line containing it to].

```
Lφ)φ G Eφ)φ]φ
```

- (b) Alteration of the first occurrence of) to] in all lines (note the effect of N which prevents a second match in a given line)

```
G Lφ)φ Eφ)φ]φ N
```

- (c) Alteration of all occurrences of) to] in all lines

```
G Lφ)φ Eφ)φ]φ
```

In examples (b) and (c) above, filename1 will be exhausted (as it will in (a) if there is no) character).

Unnecessary typing may be eliminated by use of the U command before the G command.

It is possible to cause a global loop quite easily (e.g. the command line

```
G U
```

which would be infinitely repetitive). To prevent this, EDIT allows a maximum of 40 repetitions of a command sequence on the same current line before terminating with an error message.

Matching an arbitrary string of characters

The following command allows the user, when searching for a string using F, L, A, B or E commands, to specify an arbitrary sub-string in S1.

X character	Select character, which must be a printing non-alphanumeric symbol, as an arbitrary symbol for string matching.
-------------	---

This symbol is cancelled when a new line of EDIT commands is input, or if the same symbol is used subsequently as a string delimiter. When a string S1 is being matched against the source text, an arbitrary symbol in S1 will be deemed to match a string in the source text of at least one character, up to the smallest number of characters required to generate a match with the rest of S1. Thus the commands

```
X+ L'X+Y'
```

will cause a successful match within the strings

```
XZY  
XZABCY  
ABCXABCY  
ABXABCYAB
```

but not within the strings

XY
AXYZ
AXZ
AZYB

Examples source line reads ABCDEFAB

<u>Commands</u>	<u>Results</u>
L-AA-	no match
X; L-A;A-	ABCDEF A located
X; L-A;;D-	ABCD located
X; L-A;;;D-	no match, as there are not three characters between A and D.
X; L-;D-	ABCD located
X+ L-B+E+-	BCDEF located
X+ L+B+E++	+ as delimiter cancels arbitrary effect. This is treated as L+B+ followed by a null E command.

The arbitrary symbol is treated as a normal character if it appears in string S2 of an A, B or E command.

Replacing Lines

It is possible to replace existing lines by using

R Replace the current line by the next typed line.

This command must be followed by a space or newline. The whole of the next line typed in will replace the current line, its line number being retained. Any global request including this command is cancelled.

Example To replace line 30

F30 R
The new line 30 is typed here

Inserting Lines

To insert lines of text,

I n Insert the next typed line, with line number n, after the current line.

n must be greater than the current line number and less than the next line number in the source file. Lines may be inserted before the first line of a file so long as that first line has not been altered, (ie, F,L,M,N,E,A,B or R commands have not yet been used). This new line is taken as the current line. Any global request including this command is cancelled.

Example To insert line 35 in a standard file

F30 I35

The new line 35 is typed here.

Use of the BREAK Key within EDIT

The BREAK key may be used to interrupt a global sequence at the end of a command. The message

***BROKEN

indicates that this has happened and the user must re-input the command line if he wishes to continue.

Completion of Editing

***Z

typed by itself at the beginning of a line will cause control to pass back to the command state. If delete mode is in force, the current line and the rest of filename1 will not be copied into filename2, whereas they will be if copy mode is set.

Summary of Editing Commands

A ϕ S1 ϕ S2 ϕ	insert string S2 after S1
B ϕ S1 ϕ S2 ϕ	insert S2 before S1
C	set copy mode
D	set delete mode
E ϕ S1 ϕ S2 ϕ	exchange S2 for S1
Fn	find line with line number n
F ϕ S1 ϕ	find line starting with S1
G	obey rest of command line repetitively
In	insert the next typed line, with line number n, after the current line
L ϕ S1 ϕ	locate line containing S1
Mn	miss out current line + n lines
Nn	copy current line + n lines
Pn	position character pointer at column n
R	replace the current line with the next typed line
U	unset verify mode
V	set verify mode
X character	select the character as representing an arbitrary string
***Z	end editing

Error Responses

In reply to EDIT command, these are as described under LINEDIT

After ***A

NO ARGUMENT FOR F OR L COMMAND	Illegal delimiter after F or L or no number after F
LINE TOO LONG AFTER A,B, OR E COMMAND	If the command were obeyed the current line would exceed 82 characters
MISSING DELIMITER	No delimiter in strings following A,B, or E
ERROR IN NUMBER	Typing error in number following F,M,N,P or I
GLOBAL COMMAND USED TWICE	Two G commands on the same line
UNKNOWN COMMAND	Command letter unknown (probably the user has forgotten to get out of EDIT by typing ***Z)
***BROKEN	The BREAK key has been pressed
GLOBAL LOOP	A sequence of commands has been obeyed more than 40 times for the <u>same</u> current line
FILE TOO SHORT	The last line number of the file has been passed using a N,M or R command
NO MATCH IN F OR L COMMAND	The specified string or line number cannot be found in the source text
NO MATCH IN A,B, OR E COMMAND	The specified string cannot be found in the current line
NO A,B, OR E COMMAND ALLOWED IN DELETE MODE	If D mode is selected, the current line may be lost
ILLEGAL CHARACTER AFTER X COMMAND	The next printing character after X is alphanumeric
NO SPACE OR NEWLINE AFTER DELETE COMMAND	Some other character appears immediately after D
NO SPACE OR NEWLINE AFTER REPLACE COMMAND	Some other character appears immediately after R
NUMBER OUT OF RANGE	The line number specified in I command does not lie between the current line number and the next number in the file

Examples of EDIT Commands

The following examples, annotated by the subsequent notes, are designed to illustrate some of the ways in which the EDIT command can be used. (Note the various choices of delimiters.)

TYPE ONE NUMBERED

```
20 THIS IS THE FIRST LINE IN THIS FILE
32 IT IS NUMBERED STARTING AT 20 IN STEPS OF 12
44 THIS LINE IS SHORT
56 AND HERE IS A LONG ONE CONTAINING THE STRING 'IS' TWICE
68 1234567890*:+;@[ ]<,>./
80 HERE IS THE FINAL LINE
```

EDIT ONE INTO TWO

***A

```
L/AT/ G L+IS+ P5 E#IS#WAS# N
56 AND HERE IS A LONG ONE CONTAINING THE STRING 'IS' TWICE
68 1234567890*:+;@[ ]<,>./
```

Note (i)

***Z

TYPE TWO NUMBERED

```
20 THIS IS THE FIRST LINE IN THIS FILE
32 IT IS NUMBERED STARTING AT 20 IN STEPS OF 12
44 THIS LINE WAS SHORT
56 AND HERE WAS A LONG ONE CONTAINING THE STRING 'IS' TWICE
68 1234567890*:+;@[ ]<,>./
80 HERE WAS THE FINAL LINE
```

EDIT TWO GIVING THREE

***A

```
A$LE$ NAMED THREE$
20 THIS IS THE FIRST LINE IN THIS FILE NAMED THREE
```

N 140

```
HERE IS AN INSERTION LINE WITH LINE NUMBER 40
40 HERE IS AN INSERTION LINE WITH LINE NUMBER 40
```

Note (ii)a

N M

```
56 AND HERE WAS A LONG ONE CONTAINING THE STRING 'IS' TWICE
```

F68 R

9876543210

```
68 9876543210
```

Note (ii)b

L:21: D L-TH- C

```
80 THE FINAL LINE
```

Note (iii)

***Z

10/70

TYPE THREE NUMBERED

```
20 THIS IS THE FIRST LINE IN THIS FILE NAMED THREE
32 IT IS NUMBERED STARTING AT 20 IN STEPS OF 12
40 HERE IS AN INSERTION LINE WITH LINE NUMBER 40
56 AND HERE WAS A LONG ONE CONTAINING THE STRING 'IS' TWICE
68 9876543
80 THE FINAL LINE
```

EDIT THREE, FOUR

***A

X: G L'F:E' F>F:E>JUNK

```
20 THIS IS THE JUNK IN THIS FILE NAMED THREE
20 THIS IS THE JUNK IN THIS JUNK NAMED THREE
80 THE JUNK
```

Note (iv)

***Z

TYPE FOUR NUMBERED

```
20 THIS IS THE JUNK IN THIS JUNK NAMED THREE
32 IT IS NUMBERED STARTING AT 20 IN STEPS OF 12
40 HERE IS AN INSERTION LINE WITH LINE NUMBER 40
56 AND HERE WAS A LONG ONE CONTAINING THE STRING 'IS' TWICE
68 9876543
80 THE JUNK
```

Notes

- (i) The printing of lines with numbers 56 and 68 is caused by the editor encountering the newline character after the N command. The presence of the P5 ensures that the two occurrences of the string 'THIS' and the string 'IT IS' at the beginning of lines 20, 44, and 32 respectively are not altered.
- (ii) Two methods (a) and (b) are shown for replacing existing lines. In case (a) the replacement has a different line number (40) from the original (44), whereas in case (b) both original and replacement lines have line number 68.
- (iii) Although the string '21' occurs on a different line from that containing 'TH', the newline character terminating line 68 of the file cannot be edited out in this manner.
- (iv) This example of a global edit using the X command for arbitrary string matching shows that the string 'FINAL LINE' is detected in addition to the string 'FILE'.

ENDAPPLY

Standard Format

ENDAPPLY

Purpose

To terminate the APPLY from within the specified file (see APPLY command).

Standard Format

EXAMINE

Alternative Format

EXAMINE filename

Purpose

To inspect the contents of one's own file directory either for all files (standard format) or for one particular file (alternative format).

Normal Response

Files are listed with date and time of creation, the length of the file, the access status of each user group and a count of the number of such uses (modulo 2^{16}), and the file type (TEMP = temporary, PERM = permanent).

Error Response

ERROR IN FILE TITLE	typing error in filename.
UNKNOWN FILE	file does not exist.
FILE DIRECTORY EMPTY	either no files have ever been created (no "heading") or else all have been deleted.

Standard Format

EXPAND filename STEP increment

Default Setting

increment: 10

Purpose

To add material from the console to the end of an existing file. If a key is required, it is requested in the normal manner. Write status is needed for a user to expand a file.

Normal Response

***A

signifies that all subsequent input will go into filename. The file is terminated by the user typing

***Z

by itself on a new line.

Error Responses

ERROR IN FILE TITLE	typing error in filename.
UNKNOWN FILE	file does not exist, or wrong key specified.
INSUFFICIENT STATUS	user does not have write status.
FILE BEING USED	someone else is accessing filename.
ERROR IN QUALIFIER	typing error in STEP
ERROR IN LINE NUMBER	typing error in increment
***FILE TOO LONG	a line number greater than 32767 has been allocated. The previous input line is ignored.
FILE SYSTEM FULL	all disc space has been allocated or some of the input material is lost.
PERMANENT FILE SPACE EXCEEDED - FILE GIVEN TEMP. STATUS	filename, originally selected as permanent, is too long to remain so, and has been reset as temporary.

Standard Format

FINISH

Alternative Format

LOGOUT

Purpose

To terminate a session at the console

Normal Response

FINISHED AT [time] ON [date]

Notes

A FINISH command may be forced upon an active user by the operators wishing to close down the System. In this case the normal response is suffixed by

***END OF SESSION

If a user has used more than one second of computing time on Sigma 2, the message

COMPUTING (SECS) [time in seconds]

is also output.

After logging out, ensure that the console is switched OFF at the mains (see Section 1).

Standard Format

LASTLINE filename

Purpose

To discover the last line number of filename for which the user requires at least read status.

Normal Response

The last line number of filename is typed on the user's console. If filename is empty, spaces only are typed.

Error Responses

ERROR IN FILE TITLE	typing error in filename
UNKNOWN FILE	file does not exist, or incorrect key given
INSUFFICIENT STATUS	user does not have read access to this file
FILE BEING USED	someone else is writing to this file

Standard Format

RENAME filename1 AS filename2

Purpose

To change the name of a file from filename1 to filename2. Names can only be changed for one's own files.

Normal Response

No response is given and the user is returned immediately to the command language.

Error Response

ERROR IN FILE TITLE	typing error in filename.
UNKNOWN FILE	filename1 does not exist (try using EXAMINE command).
FILE EXISTS ALREADY	filename2 exists already for another file.
ERROR IN QUALIFIER	mis-typing of AS

The following example should make clear the use of the LINEDIT facilities:

TYPE FRED NUMBERED

10 LINE TEN
20 LINE TWENTY
30 LINE THIRTY
40 LINE FORTY
50 LINE FIFTY
60 LINE SIXTY
70 LINE SEVENTY
80 LINE EIGHTY
90 LINE NINETY
100 LINE ONE HUNDRED
110 LINE ONE HUNDRED AND TEN

LINEDIT FRED GIVING FREDA

***A

5 THIS LINE IS PLACED BEFORE LINE TEN (pl)

20 THIS LINE REPLACES THE OLD LINE TWENTY (pl)

40 (nl)

60-80 (nl)

100- (nl)

***Z

TYPE FREDA NUMBERED

5 THIS LINE IS PLACED BEFORE LINE TEN
10 LINE TEN
20 THIS LINE REPLACES THE OLD LINE TWENTY
30 LINE THIRTY
50 LINE FIFTY
90 LINE NINETY

Error Responses

After the command line:

ERROR IN FILE TITLE	typing error in filename1 or filename2
UNKNOWN FILE	filename1 does not exist, or a wrong key has been given
INSUFFICIENT STATUS	user does not have read status for filename1
FILE BEING USED	someone is writing to filename1
FILE EXISTS ALREADY	filename2 has been used before
TOO MANY FILES	user already has 95 files
FILE SYSTEM FULL	there is no disc space for this file
ERROR IN QUALIFIER	typing error in GIVING or INTO

After *****A**,

ERROR IN LINE NUMBER - LINE IGNORED	typing error in line number, or numbering out of sequence
ERROR IN NUMBER TERMINATOR - LINE IGNORED	terminator is not space, newline, or -
FILE SYSTEM FULL	there is no disc space available for the rest of filename2. Some of the file is lost.
PERMANENT FILE SPACE EXCEEDED - FILE GIVEN TEMP. STATUS	filename2, originally selected as permanent, is too long to remain so and has been reset as temporary.

Standard Format

LOCALDC character

Purpose

To specify the character which will be used to denote 'delete the last character typed on this line'. The character chosen must not be a letter, a digit, a space, a newline, a comma, *, <, nor >. Any error leaves the previous setting unaltered.

Normal Response

The user is returned immediately to the command language.

Error Responses

ILLEGAL CHARACTER

one of the prohibited characters has been specified.

NO CHARACTER SPECIFIED

no character has been detected between the end of the command and the end of the line. This may be caused by attempting to repeat the current setting.

Standard Format

LOCALDL character

Purpose

To specify the character which will be used to denote 'delete the whole of the current line'. The character chosen must not be a letter, a digit, a space, a newline, a comma, *, <, nor >. Any error leaves the previous setting unaltered.

Normal Response

The user is returned immediately to the command language.

Error Responses

ILLEGAL CHARACTER

one of the prohibited characters has been specified.

NO CHARACTER SPECIFIED

no character has been detected between the end of the command and the end of the line. This may be caused by attempting to specify a character which has already been chosen as the 'delete last character' symbol.

Standard Format

LOCALTAB character

Purpose

To provide a tab facility on the DIDS display (which does not possess a tab key). This command has no effect when used on any other device. The character chosen must not be a letter, a digit, *, **nor one** of the current local editing characters. When selected, the chosen character is treated as the tab character on input, and will be converted to the appropriate number of spaces.

If, subsequent to selecting a tab character, that **same** character is selected for local editing, the local edit effect will be ignored until the tab character is altered.

The tab character is reset to null when a user logs out.

Normal Response

The user is returned immediately to the command language.

Error Response

ILLEGAL CHARACTER

one of the prohibited characters
has been specified.

LOGOUT

Format (alternative for FINISH)

LOGOUT

for details see the FINISH command.

Standard Format

LOSETABS number1 AND number2 AND

Purpose

To remove horizontal tab settings from the current list. Any number of settings may be specified and each number (which must be less than 79) is terminated either by one space, a comma, or (for the last) by newline.

Normal Response

The specified settings are cleared from the current list. If any specified setting does not appear in the current list, it is ignored. Note that position 79 may not be deleted.

Error Response

ERROR IN TAB SETTING

typing error in one of the arguments, or a setting beyond position 78 has been specified. All tab settings up to the wrong one will have been removed.

Standard Format

MESSAGE

Purpose

To type the current contents of the user's message file.

Normal Response

After typing the message file the user has the option of saving or losing its contents with the request

SAVE MESSAGE? -

The response Y or * typed on the same line (the first non-space character only is taken) will cause the file to be retained whereas any other response causes its deletion. The user is warned of the presence of a message file by the message

***MESSAGE FOR YOU

when he next enters command mode following its creation (or a subsequent insertion).

Error Response

FILE BEING USED

the message file is currently being used by someone else.

MESSAGE FILE EMPTY

no messages are waiting for this user.

Standard Format

PASSWORD

Purpose

To reset a user's password.

Normal Response

^^ (

[where ^ denotes space]

after which input of the new password is expected. Characters will not be echoed until a terminating) or newline is received.

Standard Format

```
LINEDIT filename1 GIVING filename2
```

Alternative Format

```
LINEDIT filename1 INTO filename2
```

Purpose

To create filename2 as an edited version of filename1 with the same status but null key. If a key is required for filename1, it is requested in the normal way. Read status is required for filename1.

Normal Response

***A

indicates that input can proceed. A line of input consists of a line number, followed by space, newline, or -, and possibly text terminated by a newline. If the line number corresponds to a line number in filename1, the new line from the console is taken as a replacement line. Otherwise, the console line is taken as a line to be inserted. If the line number is terminated by a space, all text up to and including the next newline is taken as the newline for filename2. If the line number is terminated by newline, that line is ignored, or, if it corresponds to a line in filename1, the effect is to delete the original line. If the line number is terminated by -, another line number or a * is expected, either of which must be terminated by newline. If it is the former, all lines between and including these will be deleted. If the latter, all lines from (and including) the first number to the end of the file are deleted.

All line numbers must be specified in ascending numerical order.

Input is terminated by the user typing

***Z

by itself on a new line.

Standard Format

RENUMBER filename FROM startnumber STEP increment

Default Settings

startnumber: 10
 increment: 10

Purpose

To change the number system of a file (for which write status is necessary). If a key is required, it is requested in the normal way.

Error Responses

ERROR IN FILE TITLE	typing error in file name.
UNKNOWN FILE	file does not exist or incorrect key specified.
INSUFFICIENT STATUS	user does not have write status for this file.
FILE BEING USED	someone else is accessing the file.
ERROR IN QUALIFIER	typing error in FROM or STEP
ERROR IN LINE NUMBER	either typing error in startnumber or increment, or the new line number is greater than 32767. In this case, the renumbering is terminated, leaving the rest of the file with the old line numbers.

Standard Format

RUN filename1 USING parameter1 AND parameter2 AND ...

Alternative Format

USING and AND are interchangeable qualifiers.

Purpose

This command enables the user to initiate an Atlas job. It assumes that filename1 commences with an Atlas job description, and variations from standard job descriptions are described later. Several conditions have to be satisfied before the job is actually sent to Atlas. In the order in which they are tested, these are:-

1. Atlas must be prepared to accept jobs;
2. the user must not have any other console initiated job still active in Atlas;
3. the user must be allowed to initiate jobs from a console (in certain circumstances he may be restrained from doing this - see later);
4. the maximum permissible number of concurrently active console initiated jobs must not be exceeded;
5. the user must have read status for filename1;
6. filename1 must not exceed 1 block in length - about 2500 characters;
7. After insertion of parameters, filename1 must start with the characters

JOB^userid^

e.g. JOB^I2000^USERNAME^AND^TITLE

The job may still be unacceptable to Atlas for the following reasons:

1. Atlas is not operating;
2. the job description contains a format error.

Both of these conditions are recognised and monitored by the System.

Normal Response

FILE INPUT TO ATLAS AT [time]

followed sometime later by

***ATLAS JOB COMPLETED

when the job has run to completion on Atlas.

Job Descriptions for Console Initiated Jobs

Acceptable job descriptions for Atlas jobs input from standard peripherals are described in 'Preparing a Complete Program for Atlas I' (ICT document CS460). Certain undocumented variations are known to be satisfactory alternatives but their use for console initiated jobs is inadvisable and they may well produce disastrous effects. The requirements and variations for console job descriptions are as follows:

1. The job title must appear on the same line as the JOB statement (see above).
2. INPUT statements are taken as referring to System files rather than documents. Thus, upon meeting a statement such as

```
INPUT 3 filename2
```

the System will translate filename2 into a private input stream (on the disc), and will amend the job description accordingly making use of filename2 as the document title. The private logical disc number used is 80 and so use of this logical number should be avoided elsewhere in the job description or in the job.

Existing private input stream definitions are recognised as such and will not cause confusion.

A maximum of 2 files may be specified as input streams.

Read status is needed for filename2 and files to which the user has only keyholder access for reading may not be used as input streams.

3. Up to two OUTPUT streams may be destined for the user's System files. The format of the statement is of the form

```
OUTPUT 2 FILE filename3
```

Again, the System amends the job description making use of one of eight private output disc areas specially reserved for the purpose, each 100 blocks long. File output on stream 0 is allocated the ultimate pseudo destination LINEPRINTER; any other stream will be apparently destined for CARDS.

Upon completion of the Atlas job, the user's file directory is scanned to see if filename3 exists. If it does, the Atlas output will be appended to the current contents of the file, irrespective of its status. However, if the file does not exist, it will be created with status FIN.P and with first line number 1. All line number increments for output will also be 1. If this causes the permanent file space allocation to be exceeded, the file will be reset as temporary, and no warning message will be given. (File statuses are described in Section 6.)

Output streams destined for System files must not be subdivided by the 'break output' facility.

The correspondence between console characters and Atlas internal code is given in Appendix I. All carriage control characters will be converted to a single newline. Any line exceeding 80 characters will have a newline inserted between the 81st and 82nd characters.

4. OUTPUT streams may also be destined for any of the normal Atlas peripherals except SEVEN HOLE PUNCH. Any output to this device will be irrevocably lost so that users requiring seven hole output should make use of their own private tape or disc and off-line it at a later stage.
5. STORE requests should be increased over normal requirements as follows:
 - (i) by 1 block, because an additional output stream (the lowest undefined one) is inserted into the job description. In this way, Sigma 2 can recognise the completion of the Atlas job, but any output to this stream generated by the job will be lost (normally it would be monitored as OUTPUT NOT DEFINED). Because of this extra output stream, the user is restricted to a maximum of 6 output streams in his job description.
 - (ii) 1 extra block of store is required per input file, because of the use of intermediate private input.

NOTE. Input and output statements in a job description referring to System files have no significance unless the job is initiated from a console.

Use of Parameters

The method of specifying these is exactly as for the APPLY command (see page 7.Apply.3), with the following additional rules.

1. Parameter substitution only takes place within the job description.
2. Any apparent error in a formal parameter does not give rise to an error message. In this case, the original file text remains unchanged.

General

Console initiated jobs must conform to the current requirements for STANDARD jobs. (The file SYSTEM/STANDARD gives details of these requirements.) Users failing to observe this restriction may, on the request of the Operations Group, be prevented from running further jobs from a console.

It is possible for jobs to be 'lost' in Atlas, such as at the time of a restart. If a user suspects that such a situation has occurred, the System operator should be informed who will check that the job has in fact been lost, and then release any pending disc areas and allow the user to re-input that job. A message

***ATLAS JOB LOST

will be typed when the areas have been released.

Output from the normal Atlas peripherals will be dealt with by the Operations Staff in the usual way.

Error Responses

COMMAND BEING USED

as there is only one buffer for information being sent down the Sigma 2 - Atlas peripheral link, the RUN command can only be entered by one user at a time.

PREVIOUS JOB STILL IN ATLAS

a user may have only one job active in Atlas.

ATLAS JOB INITIATION NOT ALLOWED

the Operations Group have requested that this user be restrained from inputting jobs to Atlas.

ATLAS JOB QUEUE FULL

input of another job would exceed the currently acceptable maximum.

UNKNOWN FILE

filename1 is not in the user's file directory, or user does not have read status.

***FILE TOO LONG

filename1 exceeds 1 block in length.

UNACCEPTABLE JOB FILE

filename1 does not start with the characters JOB userid

NO RESPONSE FROM ATLAS - JOB ABANDONED

Atlas is currently unavailable.

INPUT STOPPED BY ATLAS - INCORRECT FORMAT?

some aspect of the job description is unacceptable to Atlas. This may be caused by the appearance of Atlas document markers (***something) in filename1. In some cases, Atlas may accept such a job even though this error message is output. However, the System will have no knowledge of this job being in Atlas.

NO FILE I/O SPACE AVAILABLE

all disc areas available for input or output of files are currently in use. This may also be caused by the user requesting more than two input files or more than two output files.

JD INCOMPLETE

To be acceptable to the System, job descriptions must end with a statement such as DATA, COMPILER ..., or USE ...

NO ATLAS JOBS ALLOWED AT PRESENT

Operations have requested that Sigma 2 be inhibited from passing jobs across to Atlas.

One of the following error responses may also occur. It will be preceded by the name of the file concerned.

a) input files:-

UNKNOWN FILE

file does not exist or user only has keyholder status.

INSUFFICIENT STATUS

the user does not have read status for this file.

FILE BEING USED

someone is writing to this file.

ERROR IN FILE TITLE

typing error in file name.

b) Output files:-

ERROR IN FILE TITLE

an output stream is destined for someone else's file or a genuine typing error in file name.

Standard Format

SIGNAL userid

Purpose

To make an entry in a user's message file. The userid, SYSTEM, is available to enable system writers to receive messages from a user and the operator's console will be informed of the existence of these. The date, time, and sender of the message are automatically inserted into the message.

Normal Response

***A

indicating that input can proceed. Input cannot subsequently be edited so that use must be made of local editing facilities at input time. Input is terminated by typing

***Z

by itself at the beginning of a line.

Error Response

UNKNOWN USER

userid is not a valid user of the system.

FILE BEING USED

the message file is currently being used by someone else.

After ***A

***FILE TOO LONG

message files are restricted to 3072 characters and messages are concatenated unless cleared.

Standard Format

STATUS filename xyz.q

Purpose

To alter the degree of access of various user groups to a given file. The acceptable settings of xyz.q are given in section 6. Status settings can only be altered for one's own files.

Normal Response

No response is given and the user is returned immediately to the command language.

Error Response

ERROR IN FILE TITLE	typing error in filename.
UNKNOWN FILE	filename does not exist.
ERROR IN ARGUMENT	xyz.q is of incorrect form - existing status unaltered.
PERMANENT FILE SPACE EXCEEDED	- FILE GIVEN TEMP. STATUS filename is too long to be permanent, and so has been reset as temporary.

Standard Format

SUBMIT filename1 USING parameter1 AND parameter2 AND

Alternative Format

USING and AND are interchangeable qualifiers.

Purpose

To enable a user to run more than one job in Atlas, initiated from a console. filename1 must conform to the requirements specified with the RUN command (q.v.) but, in addition, it may not contain references to input files nor output files. The user may SUBMIT as many jobs as he likes.

Unlike the RUN command, no extra output stream for SEVEN HOLE PUNCH is inserted into the job description. Nonetheless the job should not produce output to this device as it would be directed to Sigma 2, immediately lost, and might cause confusion with another job sent by the same user to Atlas via the RUN command.

STORE requests do not need to be increased over normal requirements (contrast the RUN command).

Normal Response

FILE INPUT TO ATLAS AT [time]

There will be no information as to when or whether the job has run successfully.

Error Responses

These are the same as those for the RUN command, with the following addition:

NO I/O FILES ALLOWED

filename1 contains at least one reference to an input or an output file.

Standard Format

TABS number1 AND number2 AND

Purpose

To alter the horizontal tab settings. Any number of settings may be specified and each number (which must be less than 80) is terminated either by one space, a comma, or (for the last) by newline.

Normal Response

On subsequently using the TAB key on the console (CONTROL and I), the carriage will space up to the nearest specified TAB location and the appropriate number of spaces will be inserted in the input line. If the user is beyond the last specified TAB location, the character is translated as one space. Position 79 is always selected as a TAB location.

Error Response

ERROR IN TAB SETTING	typing error in one of the arguments, or a setting beyond position 79 has been specified. All tab settings up to the wrong one will have been accepted.
----------------------	---

Notes

Numbers should normally be specified in ascending order. However, the only requirement is that the last number specified be the largest. All positions past this number are set as TAB locations unless an error has occurred in which case only 79 will be set.

A new TABS command overwrites the old settings. (See section 5.2 concerning the use of TAB in half-duplex mode.)

If tab position 0 is set, no conversion of the tab character into spaces will take place, but the tab character will be inserted in the file, and the mechanical tab mechanism of the console (if any) will be activated when the file is TYPED. Special care must be taken since an unconverted tab will not match spaces in a string in EDIT, and Atlas will not accept the character in any input file.

Standard Format

TYPE filename FROM startnumber TO endnumber NUMBERED

Default Settings

startnumber:	first line of file
endnumber:	last line of file
NUMBERED:	no numbering

Purpose

To list a file on the user's console, in whole or in part, with or without line numbers. If the number specified in startnumber or endnumber does not correspond to a line number in the file, the nearest line numbers above and below those specified respectively are taken. If NUMBERED is specified, the line number is printed at the start of the corresponding line, separated from that line by one space. If a key is required, it is requested in the normal way.

Normal Response

Lines of the file will be listed from the specified startnumber up to (and including) endnumber.

Error Responses

ERROR IN FILE TITLE	typing error in filename.
UNKNOWN FILE	file does not exist, or incorrect key given.
ERROR IN LINE NUMBER	typing error in startnumber or endnumber.
ERROR IN QUALIFIER	typing error in FROM, TO, or NUMBERED.
INSUFFICIENT STATUS	user does not have read access to this file.
FILE BEING USED	someone else is writing to this file.
FILE TOO SHORT	startnumber is greater than the last line number of the file. This may be caused by the file being empty.

UNVERIFY

Standard Format

UNVERIFY

Purpose

To suppress the listing of the command lines during an APPLY.
See specification of APPLY for more details.

Standard Format

VERIFY

Purpose

To resume listing of the command lines during an APPLY. See specification of APPLY for more details.

Standard Format

WHO

Purpose

To list the identity and location of all users currently logged in.

Normal Response

The heading

USER

CONSOLE

is printed, underneath which is a list of user identifiers and associated console numbers of all users currently logged in. A console number with user identifier LOGIN indicates that someone is in the process of logging in at that console.

Console numbers 10 and F are normally associated with users remote from the Chilton site.

The following table lists the user identifiers of people who have had experience of using the system. Any difficulties may be resolved by SIGNALLing one of these users. Moreover, the user SYSTEM will welcome comments and criticisms of the service provided.

USERID	USER
I2011	Bart Fossey
I2081	John Baldwin
I2102	Eric Thomas
SYSTEM	Peter Blanshard, Ken Byrd

8. Suggested Use of the Multi-Access System

The following notes are intended to help a new user over the hurdle of using the System for the first time. They do not necessarily imply that there is a right or wrong way of doing a particular job nor do they exploit the System's facilities to the full.

It is assumed that the object of using the System is to develop a small program on Atlas and that the user is familiar with the conventional procedures for running Atlas jobs. (The notes are followed by an example annotated with the corresponding step number.)

- i) Obtain a 'userid' and ensure that some file space on the disc has been allocated to you.
- ii) Log in as described in section 3.
- iii) Set your password using the PASSWORD command (section 7.Password).
- iv) If the compiler needed requires the program to be in fixed format (e.g. Fortran, whose statements begin in column 7), set up appropriate tab settings with the TABS command.
- v) CREATE a file containing your program.
- vi) Ensure that the file is correct with the TYPE command (using the option NUMBERED).
- vii) If necessary, use /LINEDIT or /EDIT to make corrections to the file (see section 6 - System files).
- [viii) Use the command

TYPE SYSTEM/EDIT

to familiarise yourself with the facilities offered by the APPLY command.]

- ix) CREATE a file containing an appropriate job description, ensuring that the job parameters are within express limits (/EXPRESS lists the maxima). Output streams may be destined for files and input data may also come from files (see 7.Run).
- x) APPEND the program file to the job description.
- xi) EXAMINE the resulting file's entry in your file directory to ensure that it does not exceed 1 block in length.
- xii) If it does exceed 1 block in length then use /LINEDIT to remove the program from the job description, and EXPAND the resulting job description with an appropriate statement to cause the compiler to read the program from a different input stream, not forgetting to define the input stream in the j.d. (This step is not illustrated in the example.)

xiii) RUN the file into Atlas and log out so that other people can use the System.

xiv) Sometime later log in and see if the job has been completed and, when it has, TYPE the appropriate output files.
(If the response from Atlas is good and the job is run while you are still logged in, you should wait a few minutes before accessing any output files in order to give the System a chance to create them.)

USER

10000 ii)
()

STARTED AT 16.45.53 ON 22/07/69

PASSWORD iii)
()

TABS 7 iv)

CREATE PROGFILE v)
STATUS?
FNN.T

***A

*FORTRAN

C----- THIS IS A TRIVIAL PROGRAM TO READ 'CARD IMAGES' FROM
C----- AN INPUT STREAM (IN THIS CASE A SYSTEM FILE), AND TO
C----- OUTPUT THEM ON THE CARD PUNCH.

```
TEXT A(10)
1 READ(1,100) A
  IF(A81)-6HFINISH),99,
  WRITE(2,100) A
  GO TO 1
99 WRITE(0,101)
  GO TO EXIT
100 FORMAT(10A8)
101 FORMAT(13H0JOB COMPLETE)
END
```

*DONTMAP

*ENTER

***Z

TYPE PROGFILE NUMBERED vi)

```
10 *FORTRAN
20 C----- THIS IS A TRIVIAL PROGRAM TO READ 'CARD IMAGES' FROM
30 C----- AN INPUT STREAM (IN THIS CASE A SYSTEM FILE), AND TO
40 C----- OUTPUT THEM ON THE CARD PUNCH.
50 TEXT A(10)
60 1 READ(1,100) A
70 IF(A81)-6HFINISH),99,
80 WRITE(2,100) A
90 GO TO 1
100 99 WRITE(0,101)
110 GO TO EXIT
120 100 FORMAT(10A8)
130 101 FORMAT(13H0JOB COMPLETE)
140 END
150 *DONTMAP
160 *ENTER
```

/EDIT,PROGFILE

.vii)

***A

F70

70 IF(A81)-6HFINISH),99,

E'A81'A(1'

70 IF(A(1)-6HFINISH),99,

***Z

TYPE /EDIT
UNVERIFY
EDIT <1>,XJUNKX
STATUS <1> FNN.T
DELETE <1>
RENAME XJUNKX AS <1>
ENDAPPLY

viii)

CREATE MYDATA
STATUS?
FNN.T

***A

THIS IS THE FIRST OF MY CARDS TO BE LISTED
HERE IS ANOTHER
AND HERE IS THE LAST
FINISH

***Z

CREATE JDFILE
STATUS?

ix)

STANDARD STATUS ANN.T ASSUMED

***A

JOB 1000 MYNAME AND TITLE
COMPUTING 5 SECONDS
INPUT 1 MYDATA
OUTPUT 0 FILE XYZ1
OUTPUTM- 2 CARDS 25 LINES
USE HARTRAN

***Z

APPEND PROGFILE TO JDFILE

x)

EXAMINE JDFILE

xi)

JDFILE 17.01 22/07 1 A 2 N - N - TEMP

RUN JDFILE
FILE INPUT TO ATLAS AT 17.03.19

xiii)

LOGOUT

FINISHED AT 17.03.24 ON 22/07/69
COMPUTING (SECS) 4

7/69

8.3

USER
I0000
()

STARTED AT 17.04.48 ON 22/07/69

xiv)

***ATLAS JOB COMPLETED

TYPE W-XYZ1
OUTPUT 0
I0000 MYNAME AND TITLE

22/07/69 17.03.29
VERSION 7 HARTRAN SYSTEM DATED 20 MAY 1968

OUTPUT 15 UNDEFINED - ST
***BREAK? - YES

LASTLINE XYZ1
107

TYPE XYZ1 FROM 73

JOB COMPLETE
END OF JOB.

TAPE NUMBER	POSITIONED AT BLOCK	WORD
80	2	0
110	1	0

I0000 MYNAME AND TITLE
DATE 22.07.69
TIME 17.03.42
SERIAL NUMBER 8970905

	REQUESTED	USED	COMPILE
INSTRUCTION INTERRUPTS	800	159	149
COMPILE STORE	120	119	
EXECUTION STORE	3		

***BREAK? - YES

DELETE XYZ1
FILE DELETED

DELETE PROGFILE
FILE DELETED

DELETE JDFILE
INSUFFICIENT STATUS

STATUS JDFILE FNN.T

DELETE JDFILE
FILE DELETED

LOGOUT

FINISHED AT 17.08.00 ON 22/07/69
COMPUTING (SECS) 1

Appendix I

Atlas Internal Code equivalents of Console Characters

The following table gives the correspondence between console characters and Atlas Internal Code (AIC). Certain equivalences are only meaningful in the output sense. Undefined AIC characters are ignored on output and all carriage control characters will be converted into a single newline.

<u>AIC (octal)</u>	<u>Console Character</u>	<u>AIC (octal)</u>	<u>Console Character</u>
Inner Set		Outer Set	
01	space	01	space (output only)
02	space (output only)	03	;
03	←	17	:
10	(21	[
11)	22]
12	,	23	→
13	⊥	25	≠
14	?	26	↑
15	^	27	
16	*	30	↓
17	/	31	~
20-31	0-9	32	10
32	<	41-72	a-z (output only; printed as upper case on console)
33	>		
34	=		
35	+		
36	-		
37	.		
40	'		
41-72	A-Z		

